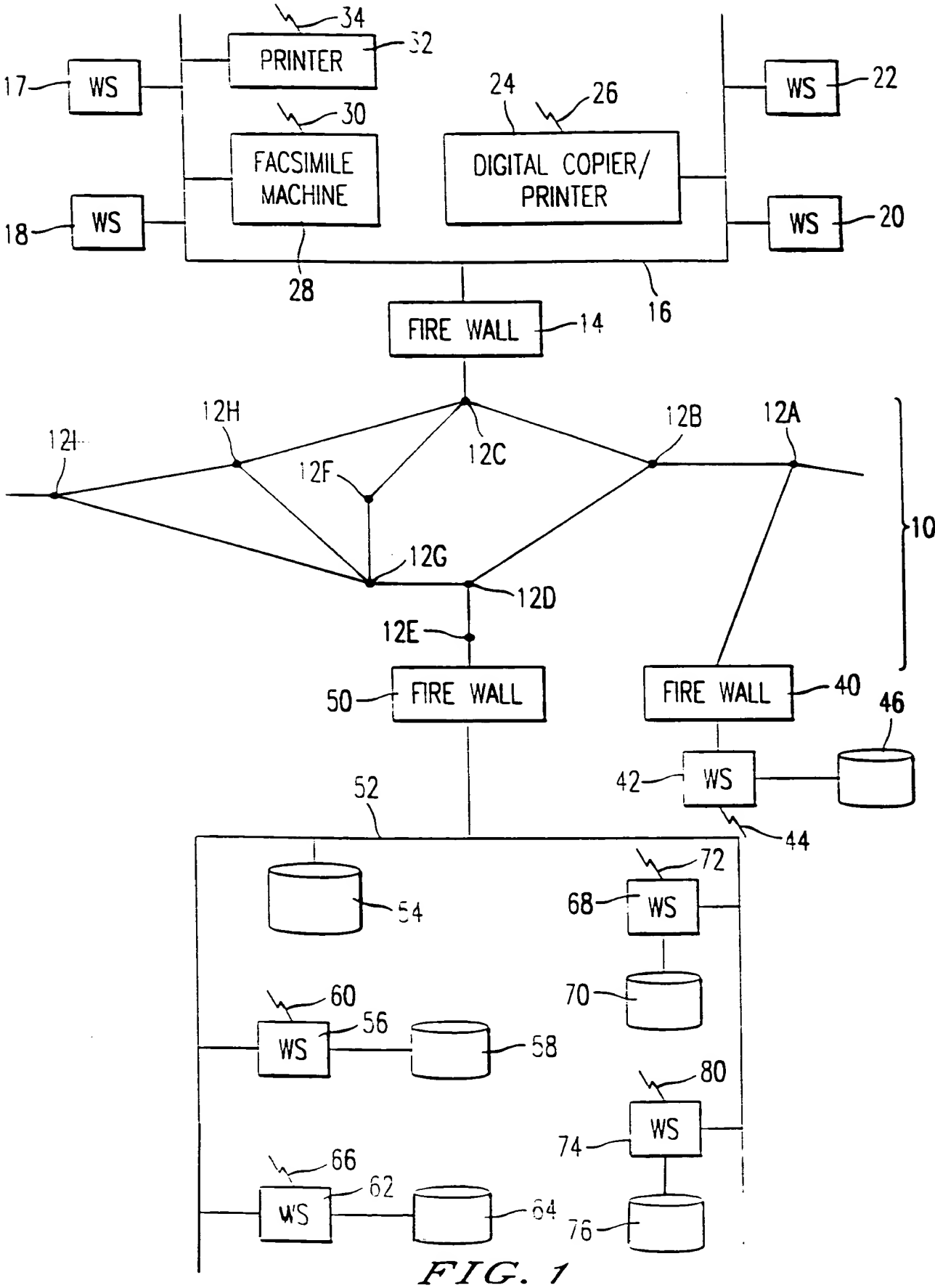

34
PRINTER
32
17
WS
24
26
WS
22
30
FACSIMILE MACHINE
DIGITAL COPIER/ PRINTER
18
WS
WS
20
28
16
FIRE WALL
14
12H
12B
12A
12I
12C
12F
10
12G
12D
12E
50
FIRE WALL
FIRE WALL
40
46
42
WS
52
44
72
54
68
WS
70
60
56
WS
58
80
WS
74
66
62
WS
64
76

FIG. 1

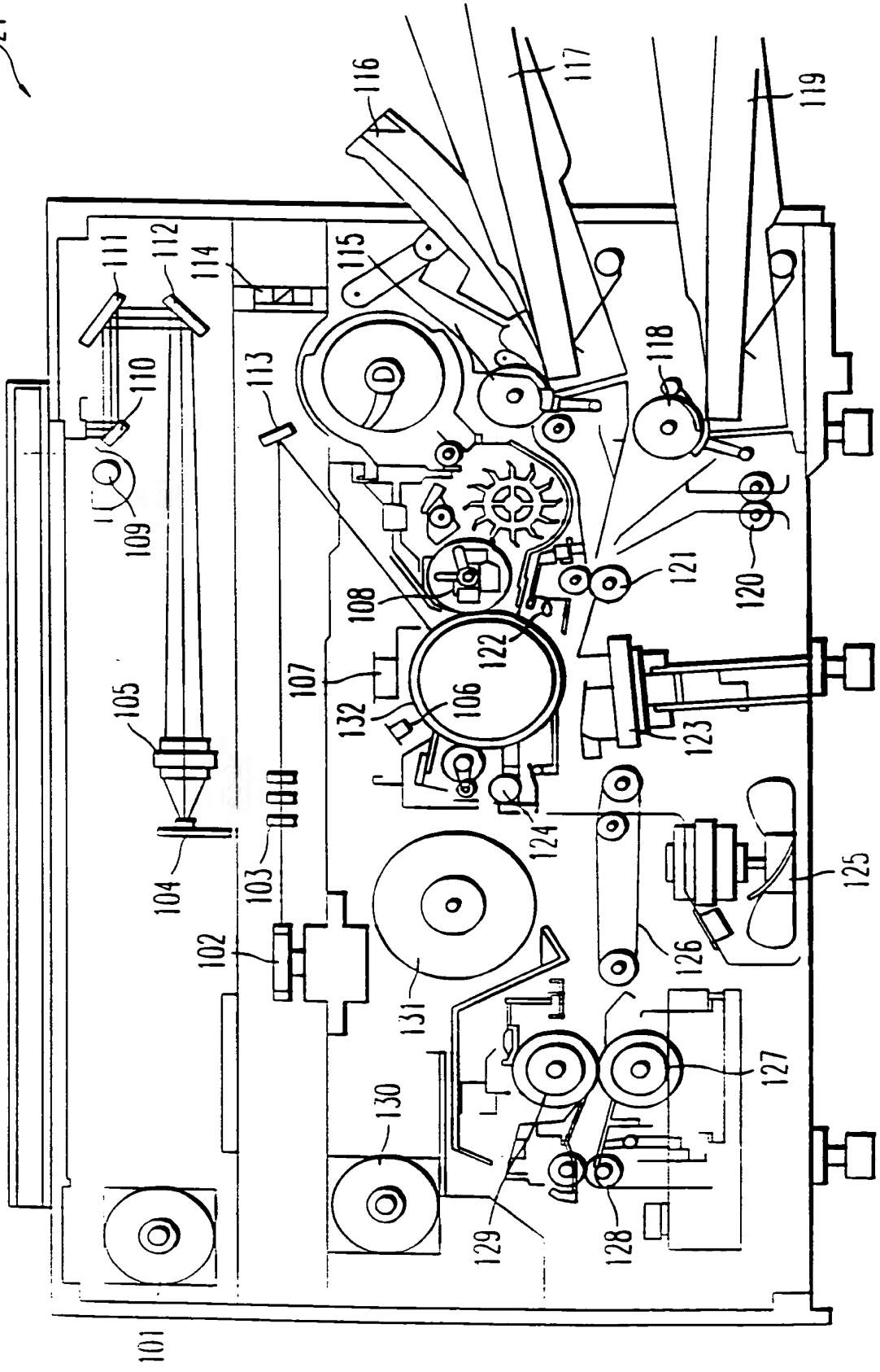

24
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

FIG. 2

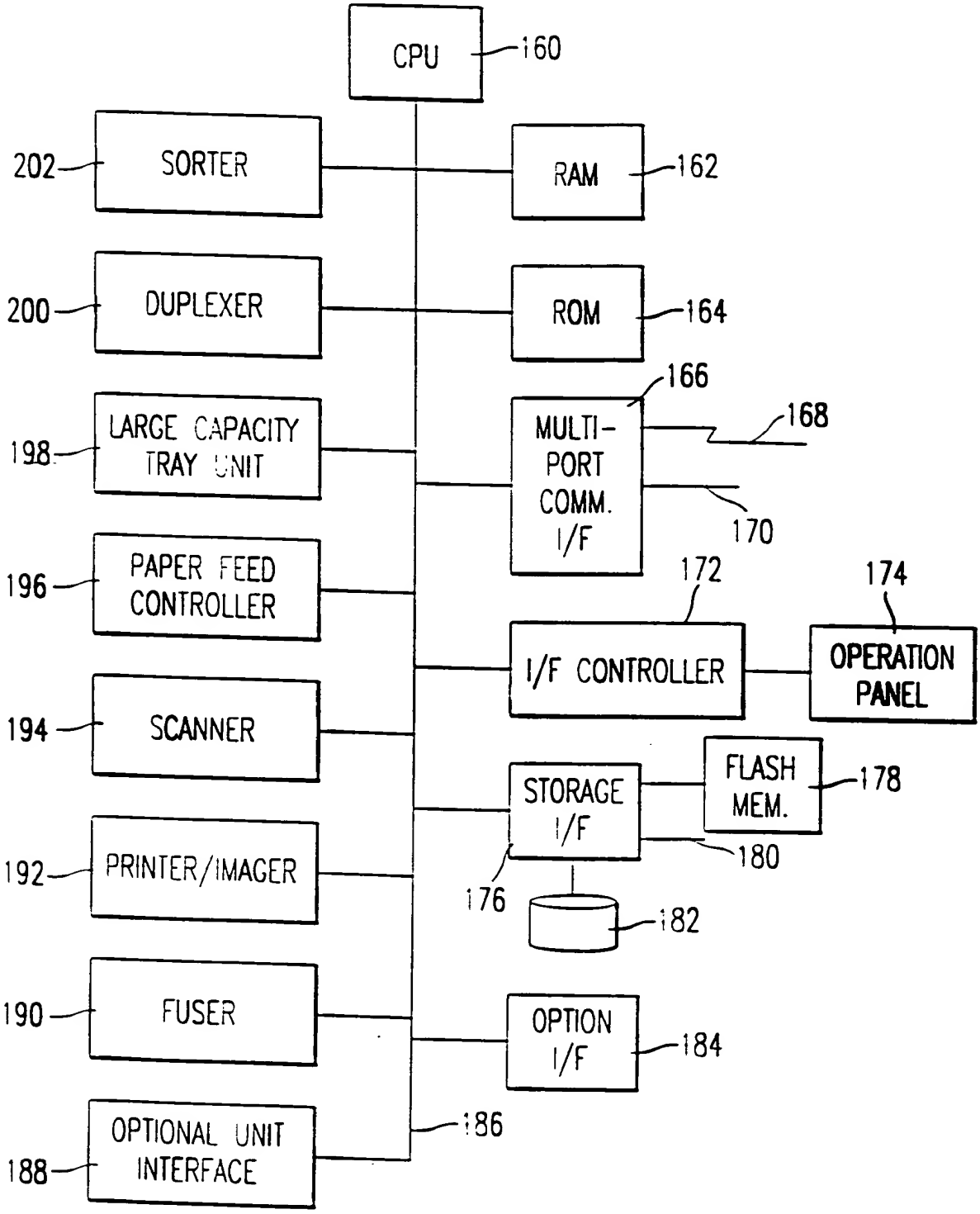

CPU
160
SORTER
202
RAM
162
DUPLEXER
200
ROM
164
LARGE CAPACITY TRAY UNIT
198
MULTI-PORT COMM. I/F
166
168
170
PAPER FEED CONTROLLER
196
I/F CONTROLLER
172
OPERATION PANEL
174
SCANNER
194
STORAGE I/F
FLASH MEM.
178
180
PRINTER/IMAGER
192
176
182
FUSER
190
OPTION I/F
184
OPTIONAL UNIT INTERFACE
188
186

FIG. 3

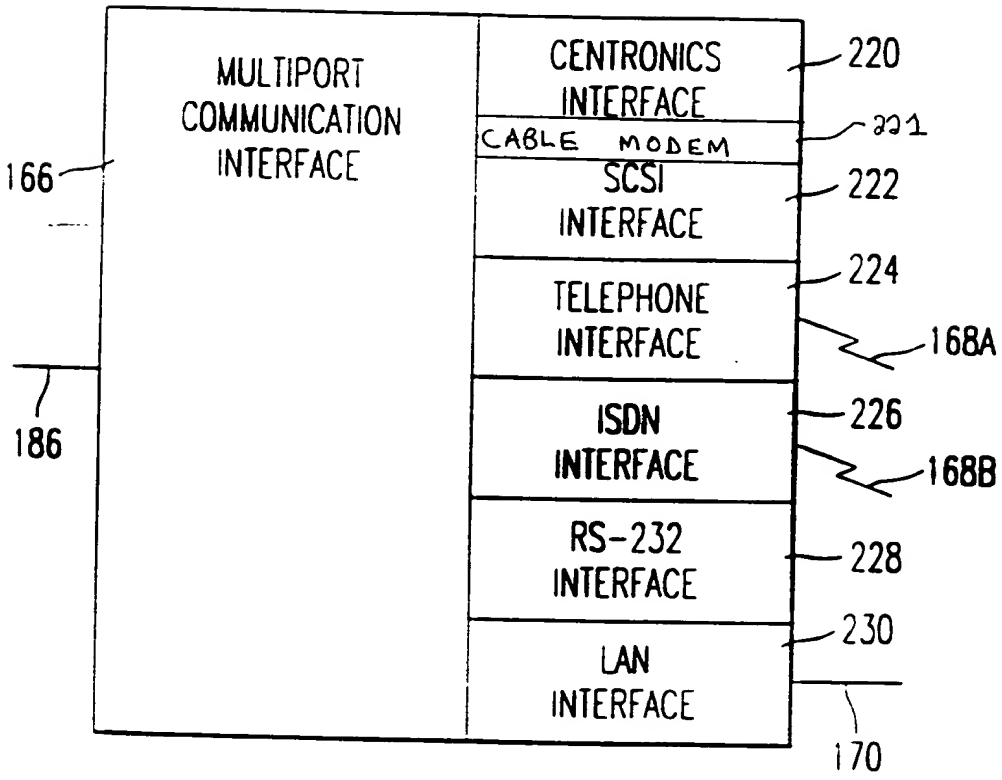

MULTIPORT
COMMUNICATION
INTERFACE
166
186
CENTRONICS
INTERFACE
220
CABLE MODEM
221
SCSI
INTERFACE
222
TELEPHONE
INTERFACE
224
168A
ISDN
INTERFACE
226
168B
RS-232
INTERFACE
228
LAN
INTERFACE
230
170

FIG. 4

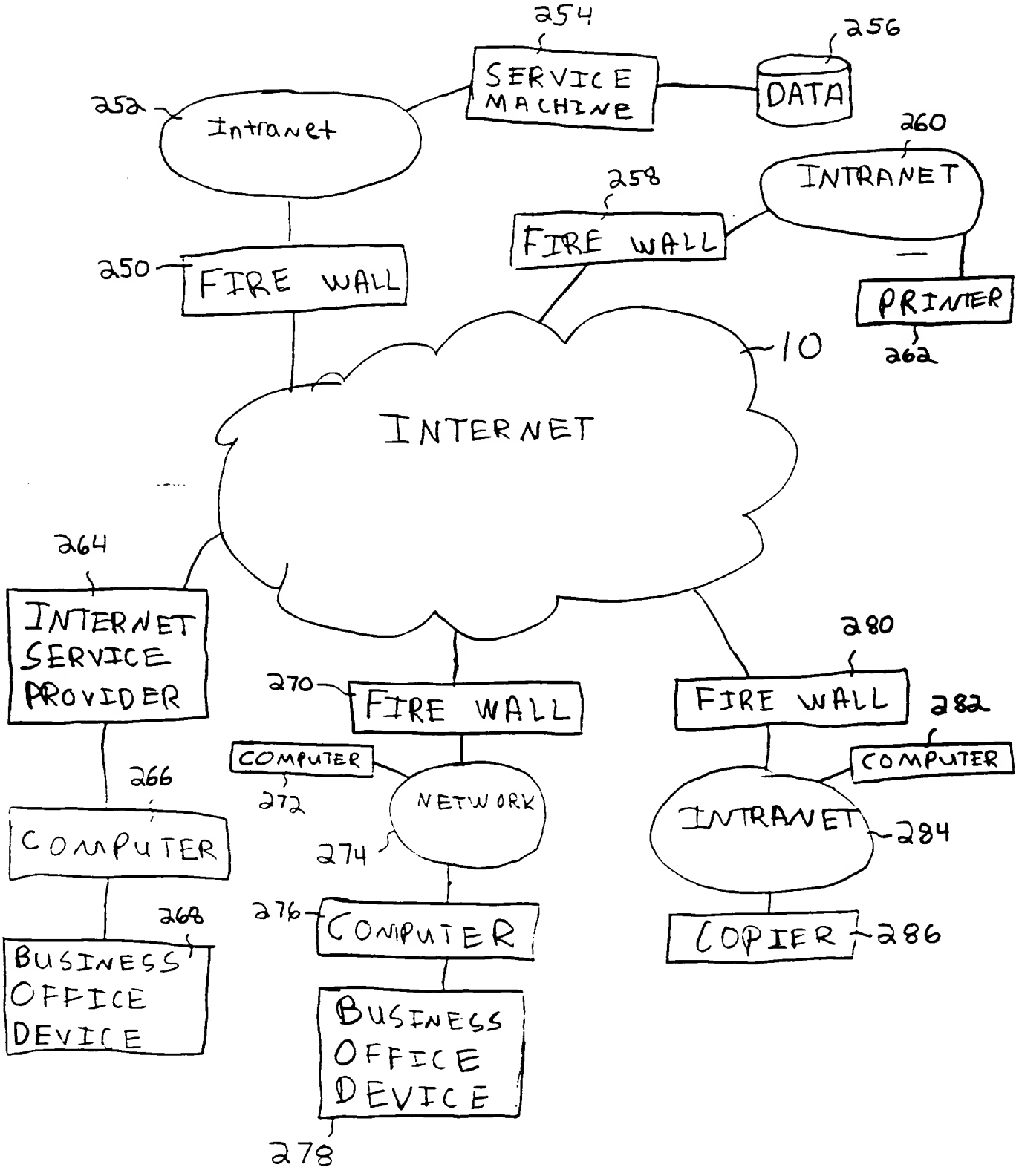

254
SERVICE MACHINE
256
DATA
252
Intranet
260
INTRANET
258
FIRE WALL
250
FIRE WALL
PRINTER
262
10
INTERNET
264
INTERNET SERVICE PROVIDER
270
FIRE WALL
280
FIRE WALL
282
COMPUTER
COMPUTER
272
NETWORK
266
COMPUTER
274
INTRANET
284
276
COMPUTER
COPIER
286
268
BUSINESS OFFICE DEVICE
BUSINESS OFFICE DEVICE
278

Fig. 5

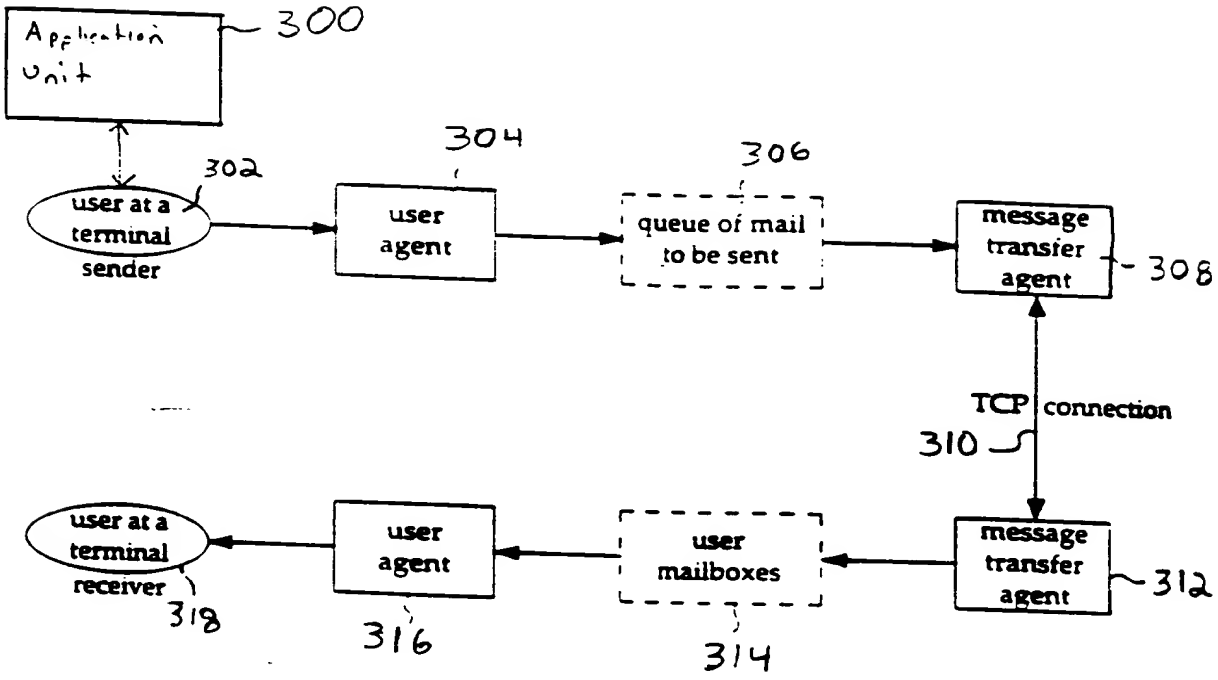

Application unit
300
user at a terminal sender
302
user agent
304
queue of mail to be sent
306
message transfer agent
308
TCP connection
310
user at a terminal receiver
318
user agent
316
user mailboxes
314
message transfer agent
312

Fig. 6A

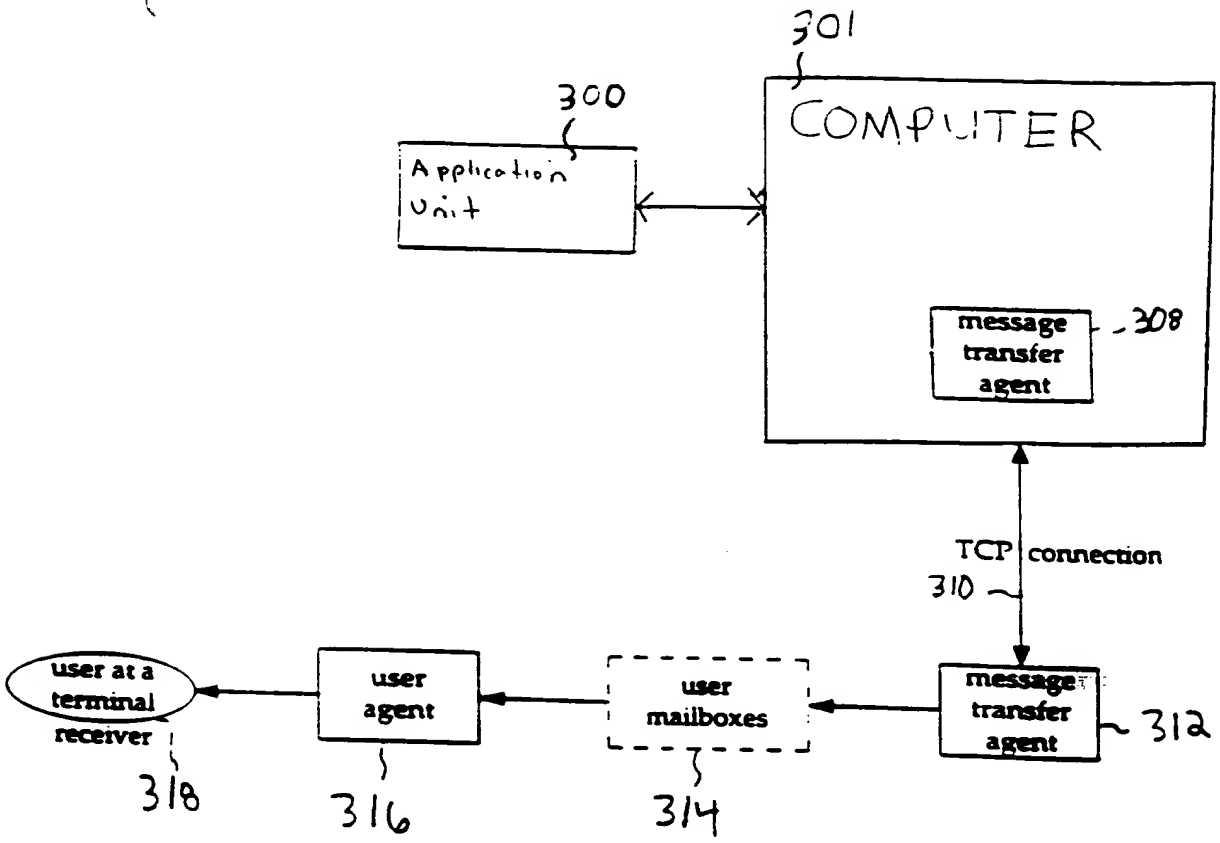

301
300
COMPUTER
Application Unit
message transfer agent
308
TCP connection
310
user at a terminal receiver
user agent
user mailboxes
message transfer agent
312
318
316
314

Fig. 6B

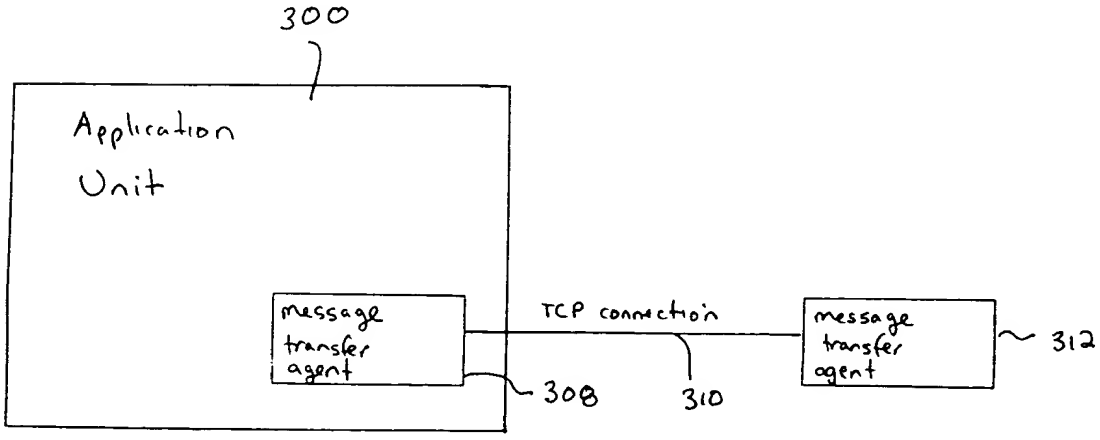

300
Application Unit
message transfer agent
TCP connection
message transfer agent
312
308
310

FIG. 6C

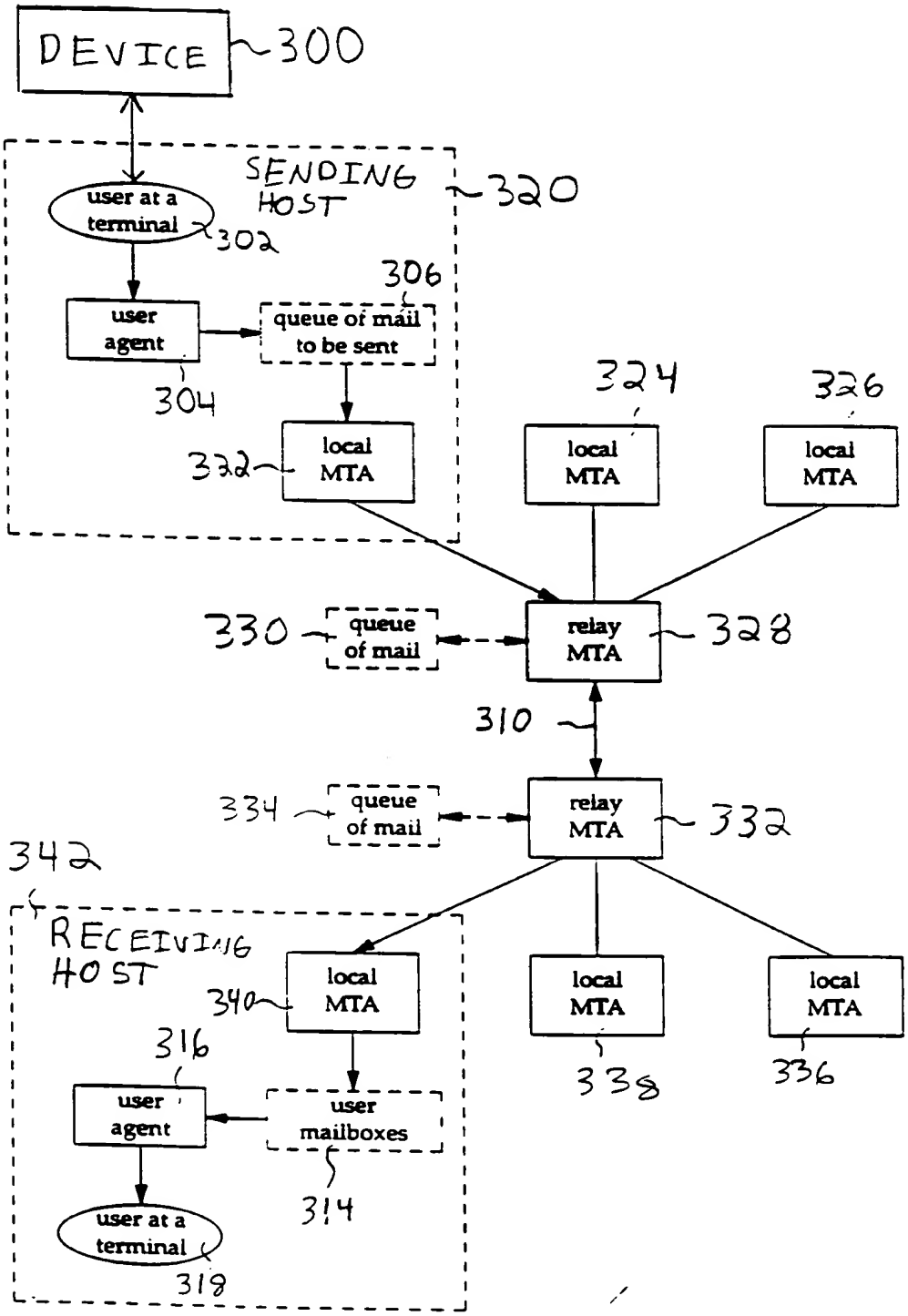

DEVICE
300
SENDING HOST
320
user at a terminal
302
306
user agent
queue of mail to be sent
304
local MTA
322
324
local MTA
326
local MTA
330
queue of mail
relay MTA
328
310
334
queue of mail
relay MTA
332
342
RECEIVING HOST
local MTA
340
316
user agent
user mailboxes
314
user at a terminal
318
local MTA
338
local MTA
336

Fig. 7

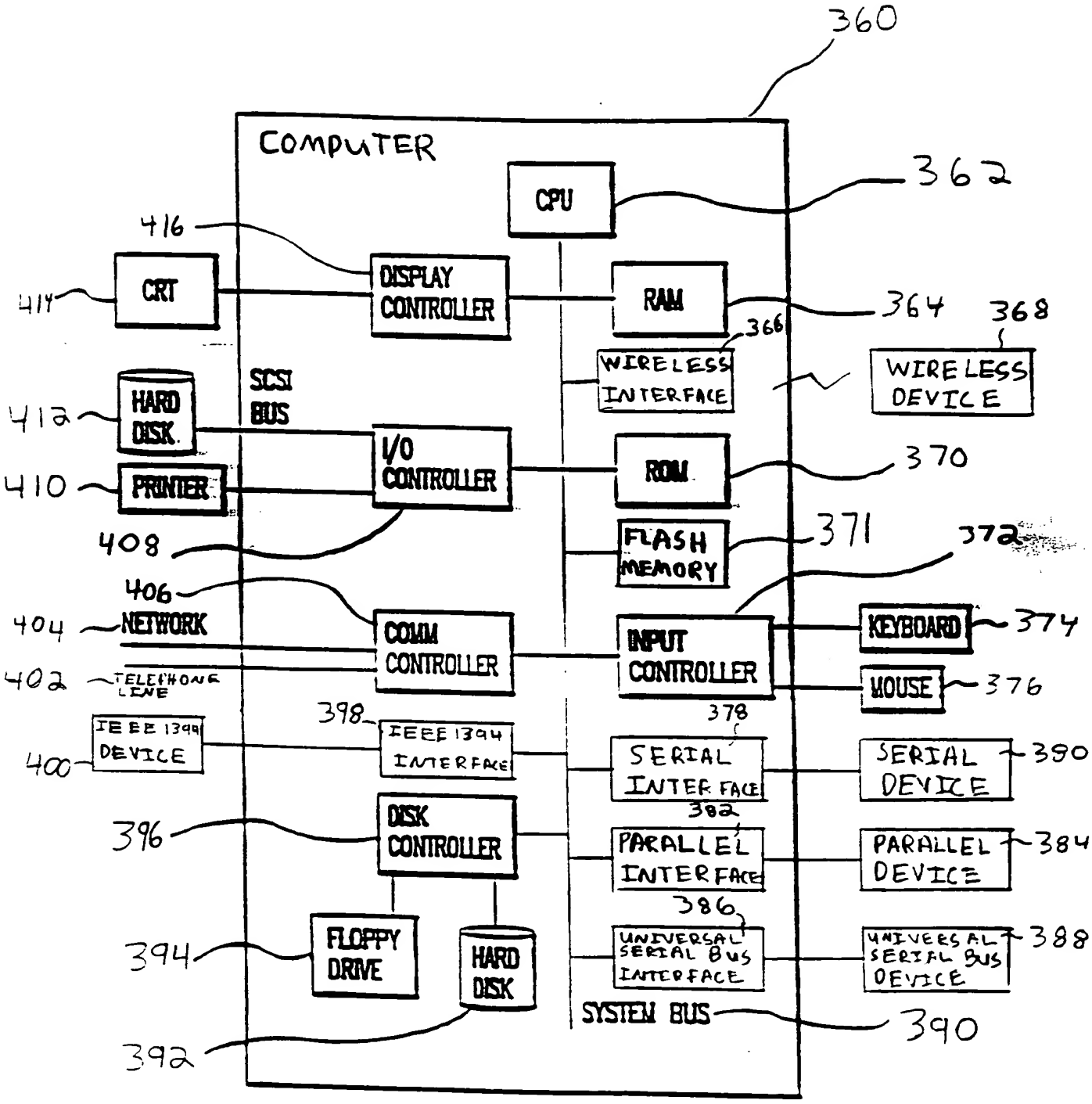

360
COMPUTER
CPU
362
416
DISPLAY CONTROLLER
414
CRT
RAM
364
368
366
WIRELESS INTERFACE
WIRELESS DEVICE
SCSI BUS
412
HARD DISK
I/O CONTROLLER
ROM
370
410
PRINTER
408
FLASH MEMORY
371
372
406
404
NETWORK
COMM CONTROLLER
INPUT CONTROLLER
KEYBOARD
374
402
TELEPHONE LINE
MOUSE
376
398
378
IEEE 1394 DEVICE
IEEE 1394 INTERFACE
SERIAL INTERFACE
SERIAL DEVICE
390
400
382
396
DISK CONTROLLER
PARALLEL INTERFACE
PARALLEL DEVICE
384
386
394
FLOPPY DRIVE
HARD DISK
UNIVERSAL SERIAL BUS INTERFACE
UNIVERSAL SERIAL BUS DEVICE
388
SYSTEM BUS
390
392

Fig. 8

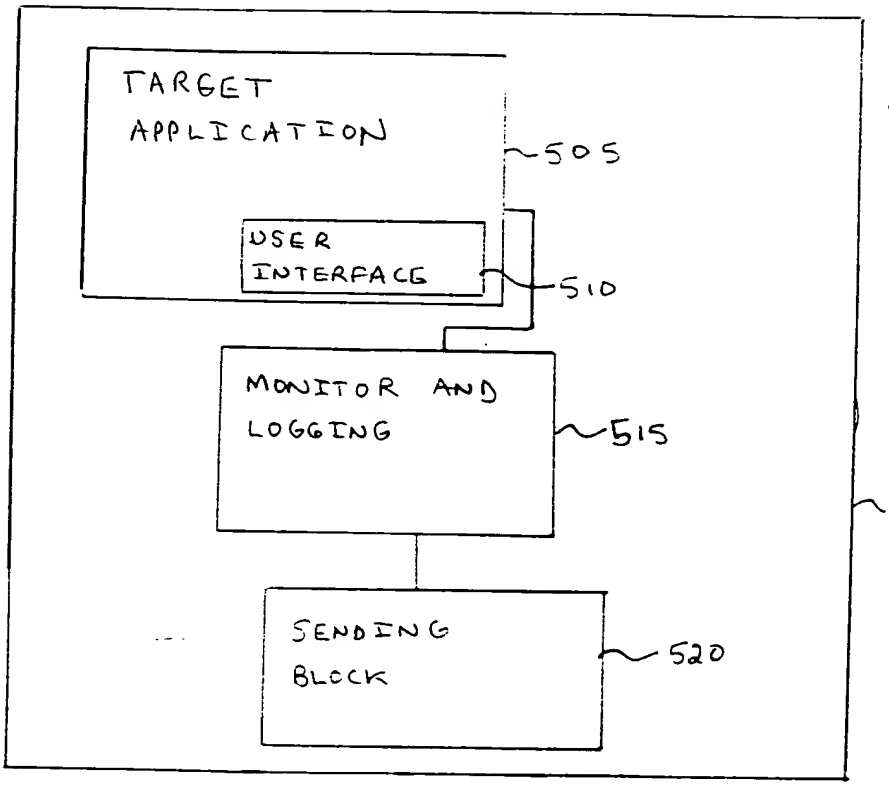


FIG. 9

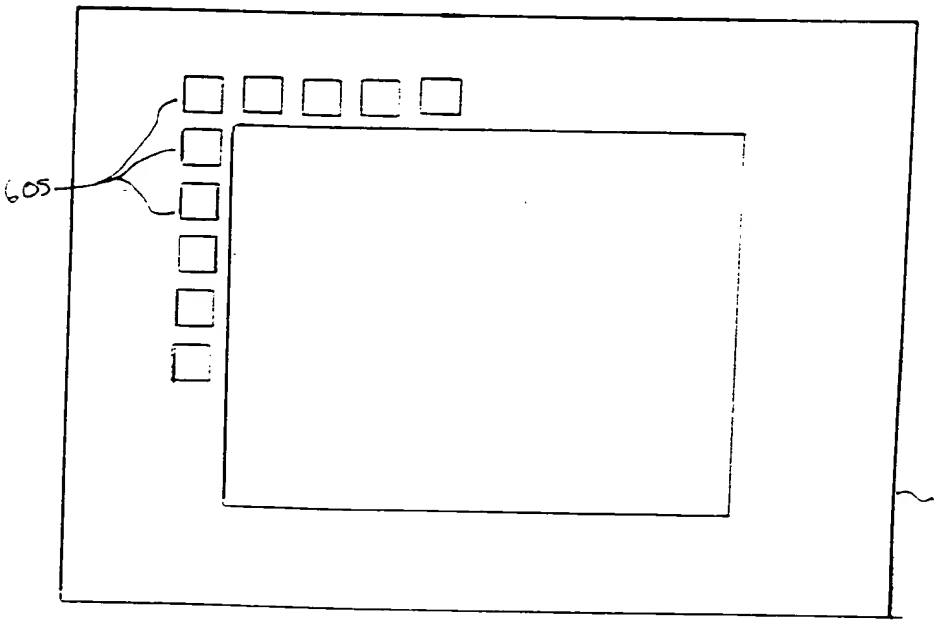
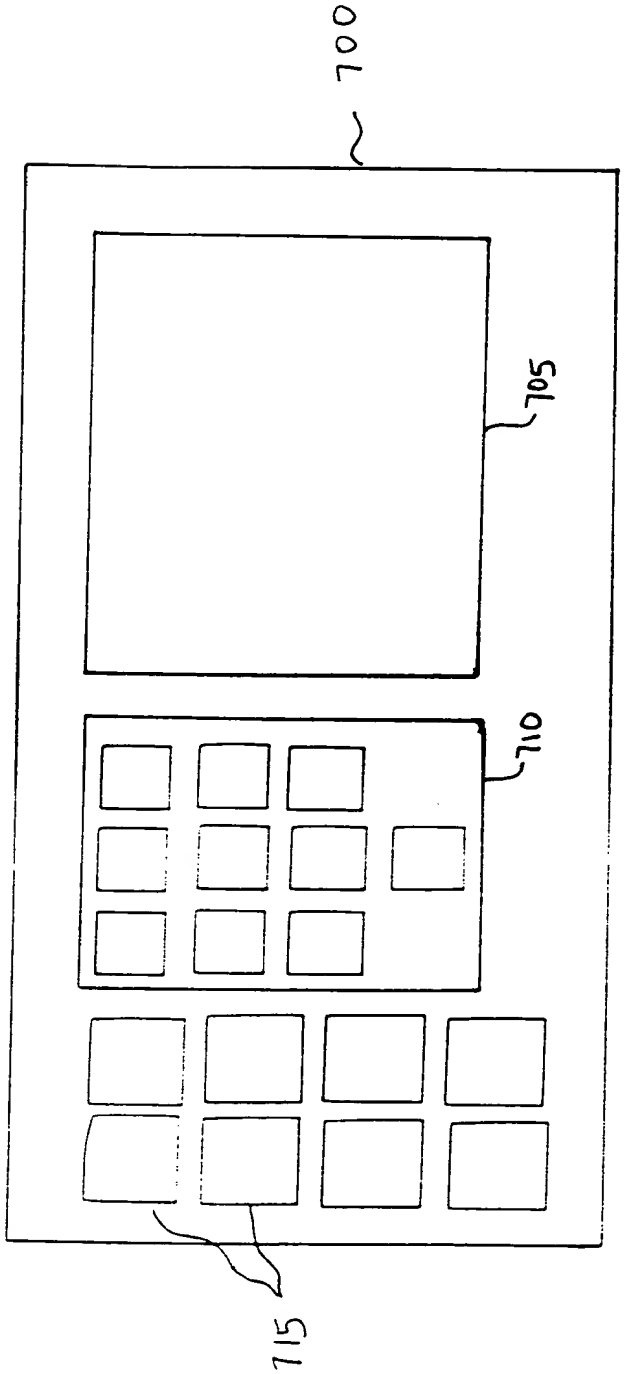


FIG. 10

09/750,958 - 12/28/00

FIG. 11



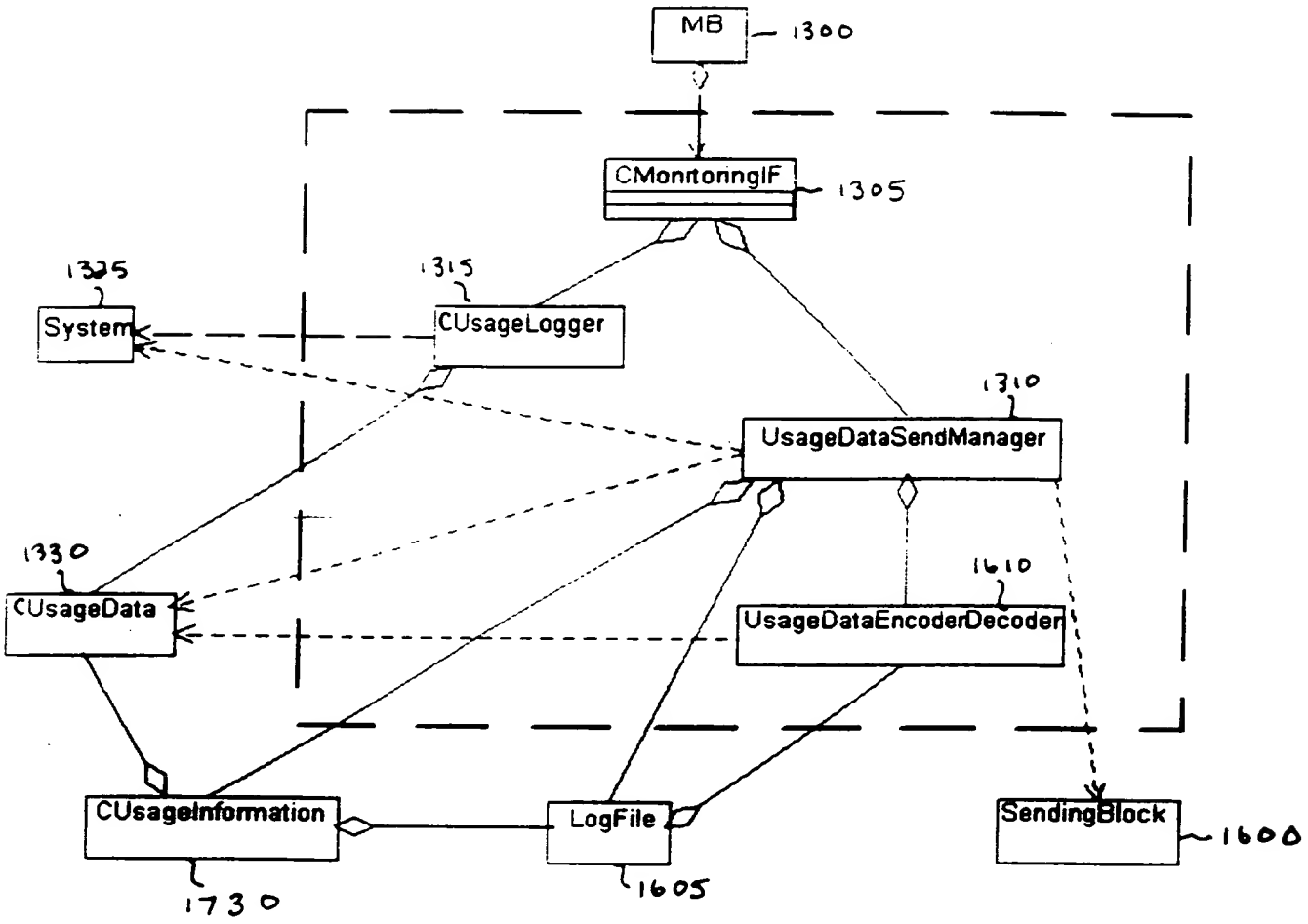

MB
1300
CMonitoringIF
1305
1325
System
1315
CUsageLogger
1310
UsageDataSendManager
1330
CUsageData
1610
UsageDataEncoderDecoder
CUsageInformation
1730
LogFile
1605
SendingBlock
1600

FIG. 12

StartMonitoring()

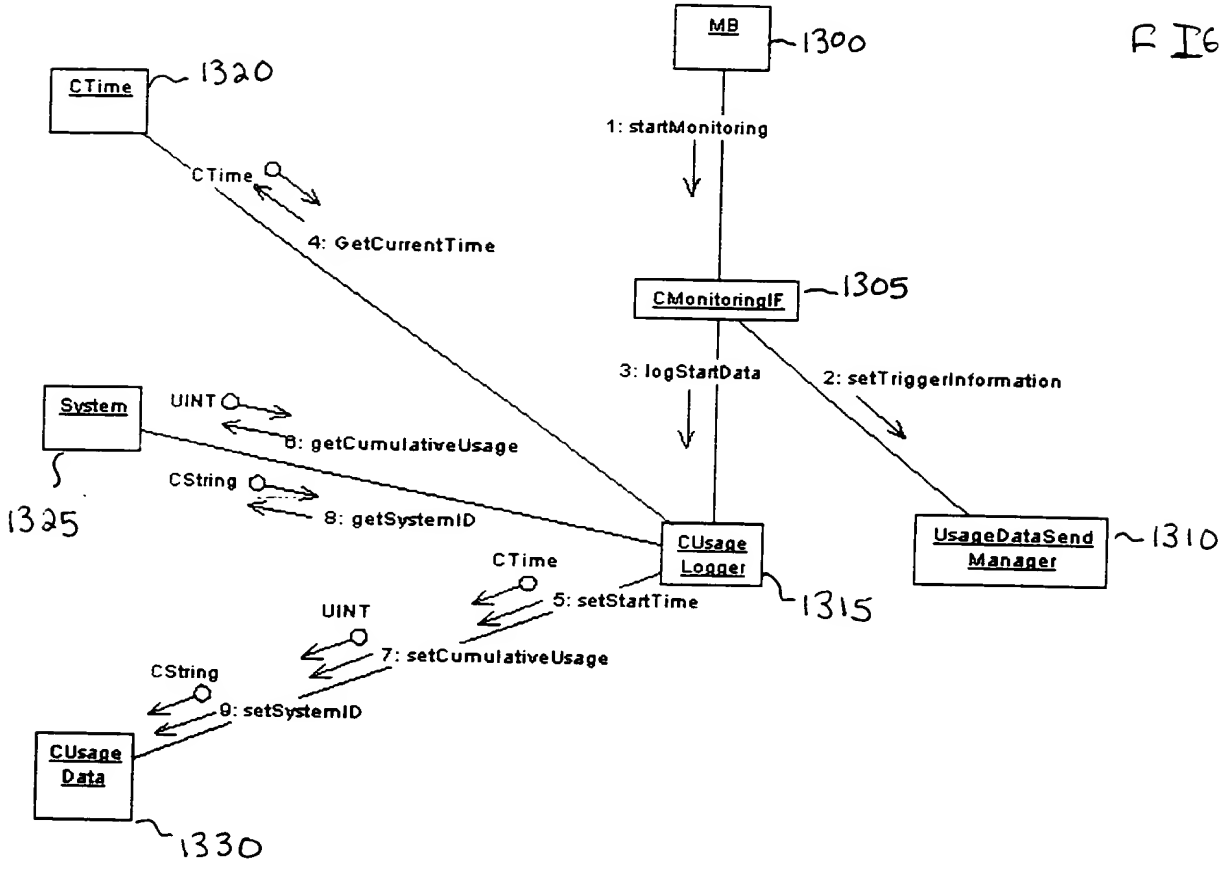
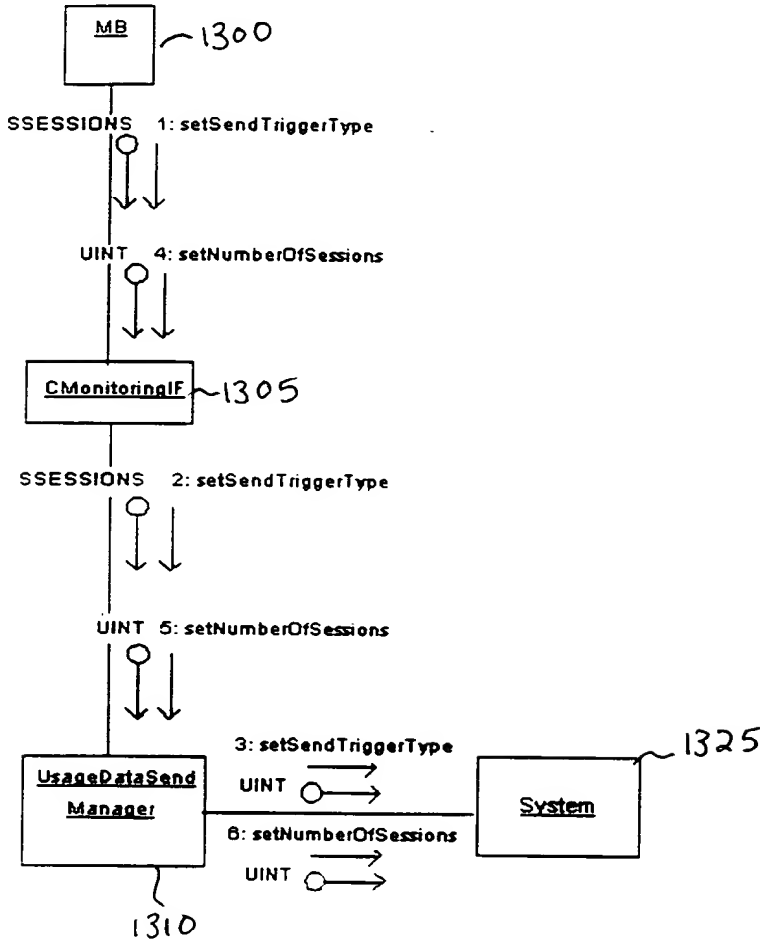


FIG. 13

setSendTriggerType()
&
setNumberOfSessions()

Fig. 14



CommandUsage() Scenario

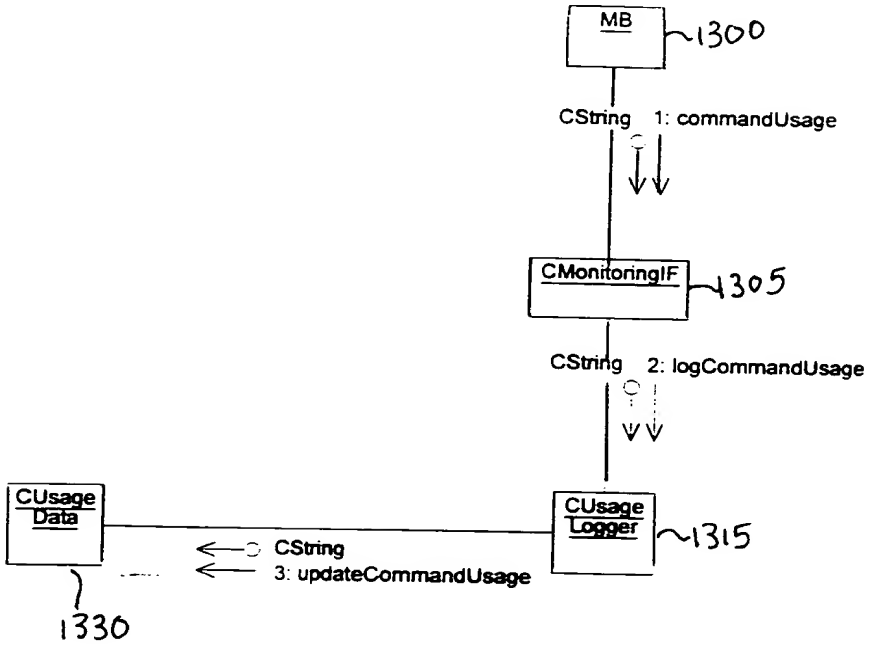


Fig. 15A

CommandUsage() Scenario

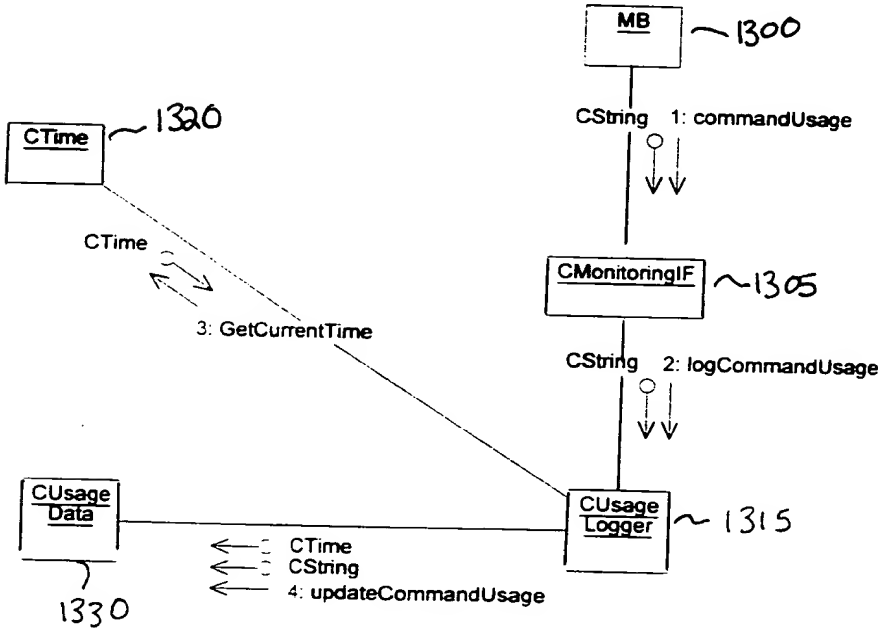
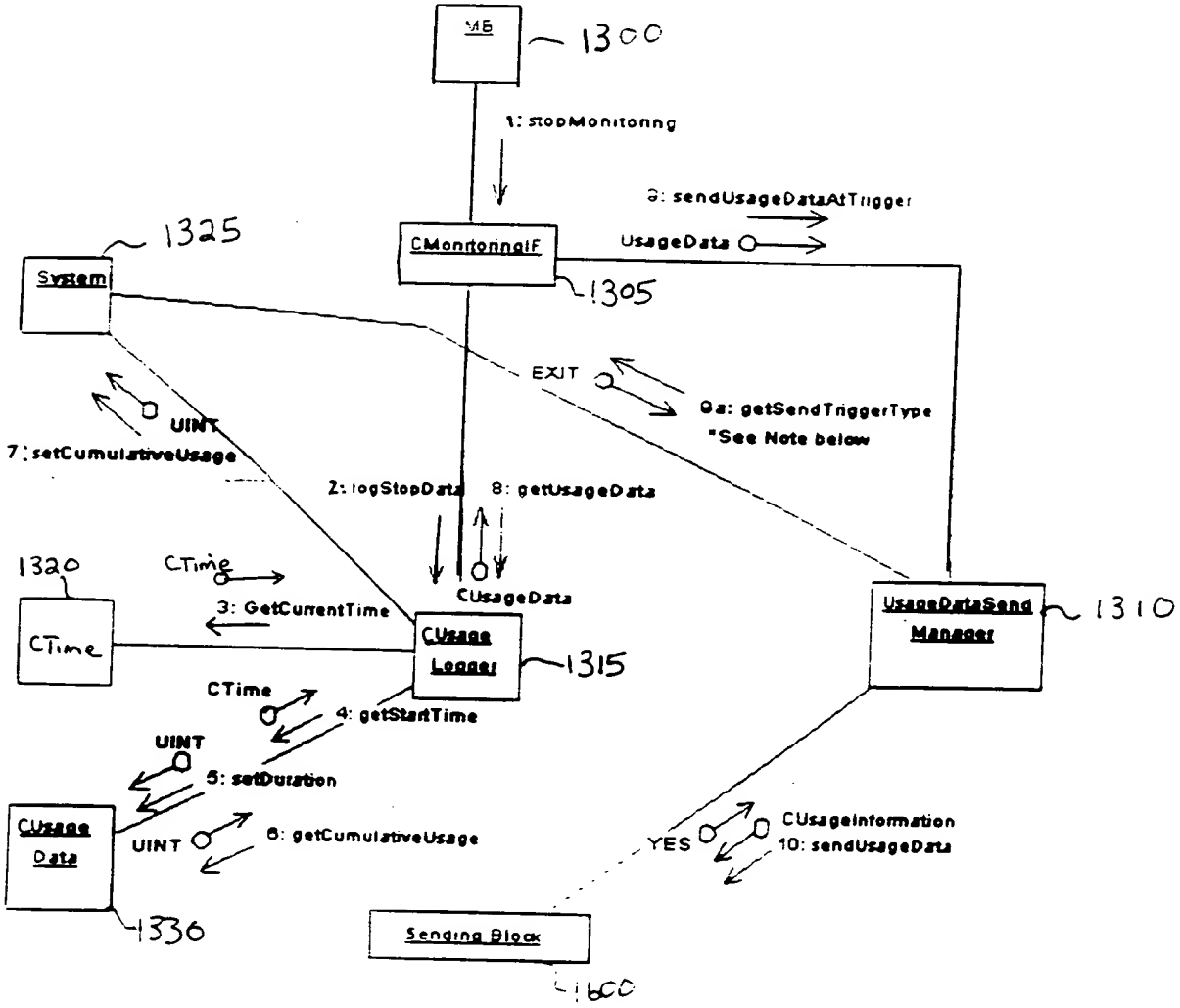


Fig. 15B

stopMonitoring()

Send Trigger on Exit immediately

Fig. 16A



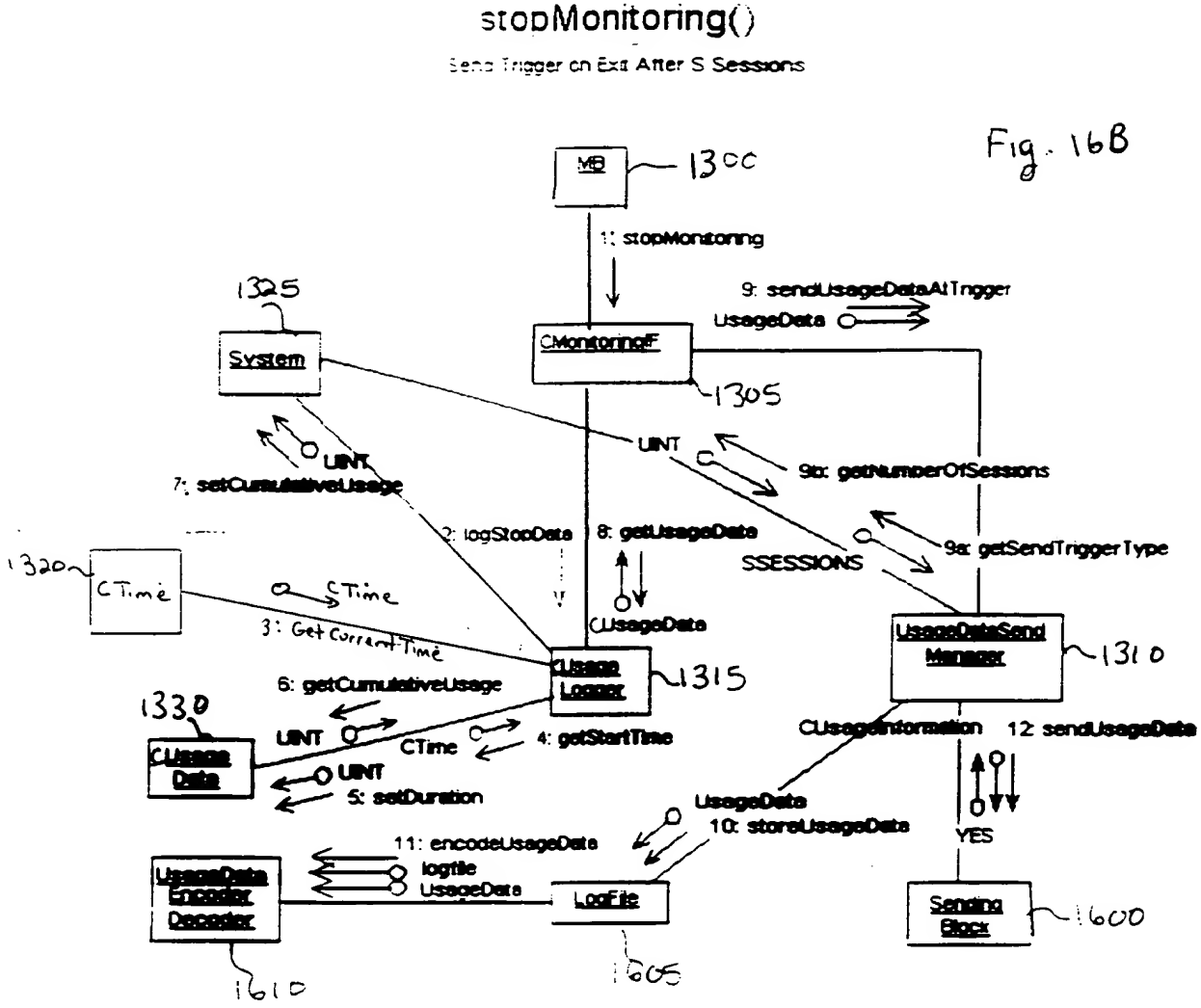
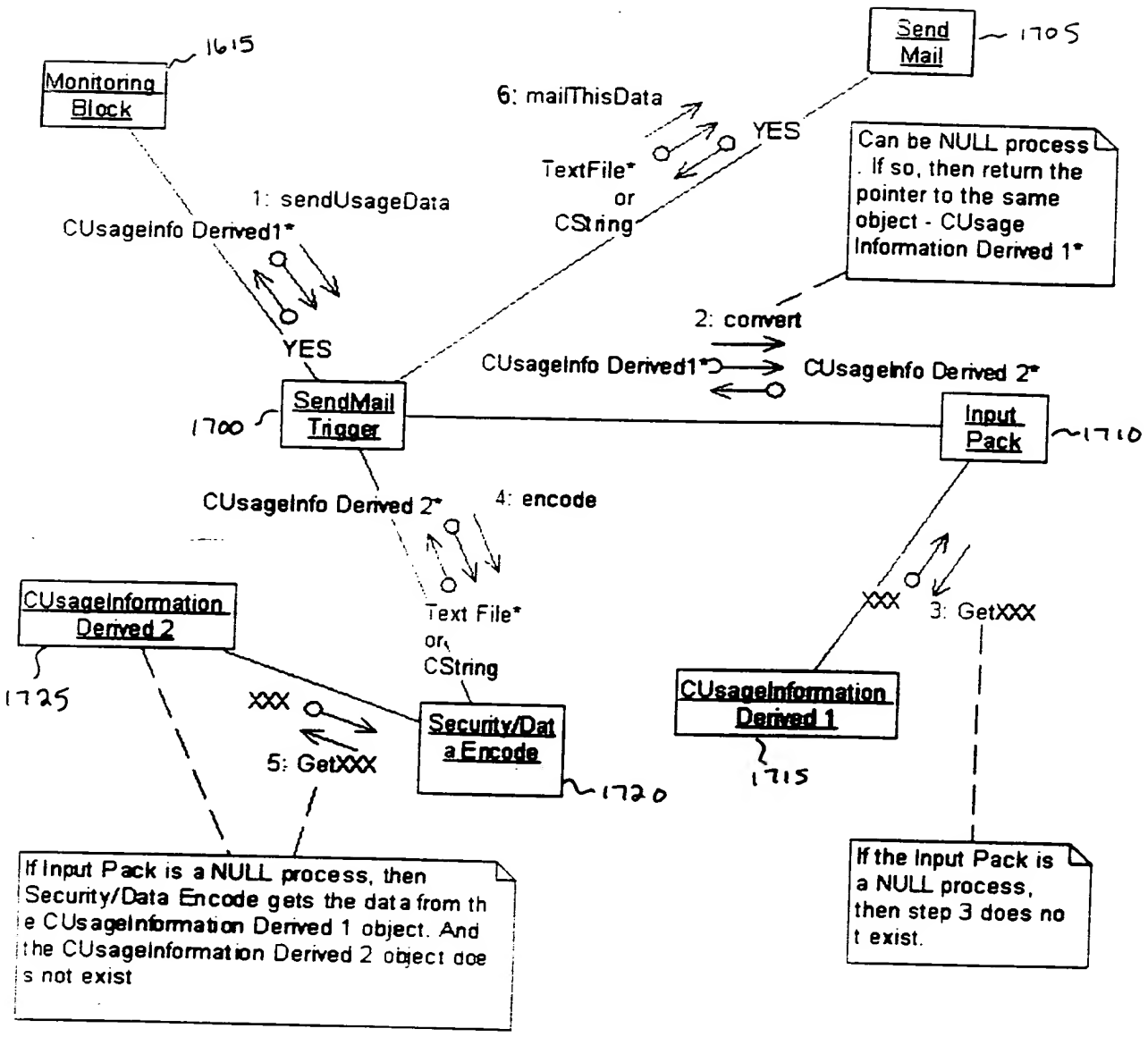

stopMonitoring()
Send Trigger on Exit After S Sessions
MB
1300
1: stopMonitoring
1325
System
CMonitoringF
1305
9: sendUsageDataAtTrigger
UsageData
UINT
7: setCumulativeUsage
2: logStopData
8: getUsageData
9b: getNumberOfSessions
SSESSIONS
9a: getSendTriggerType
1320
CTime
STime
3: GetCurrentTime
CUsageData
UsageDataSend Manager
1310
CUsage Logger
1315
6: getCumulativeUsage
1330
CUsage Data
UINT
CTime
4: getStartTime
UINT
5: setDuration
CUsageInformation
12: sendUsageData
UsageData
10: storeUsageData
YES
11: encodeUsageData
logfile
UsageData
UsageData Encoder Decoder
LogFile
Sending Block
1600
1610
1605

Fig. 16B

Fig. 17


Monitoring Block
1615
Send Mail
1705
6: mailThisData
YES
TextFile*
or
CString
Can be NULL process . If so, then return the pointer to the same object - CUsage Information Derived 1*
1: sendUsageData
CUsageInfo Derived1*
YES
2: convert
CUsageInfo Derived1*
CUsageInfo Derived 2*
SendMail Trigger
1700
Input Pack
1710
CUsageInfo Derived 2*
4: encode
CUsageInformation Derived 2
1725
Text File*
or
CString
XXX
3: GetXXX
CUsageInformation Derived 1
1715
Security/Data Encode
1720
5: GetXXX
If Input Pack is a NULL process, then Security/Data Encode gets the data from the CUsageInformation Derived 1 object. And the CUsageInformation Derived 2 object does not exist
If the Input Pack is a NULL process, then step 3 does not exist.